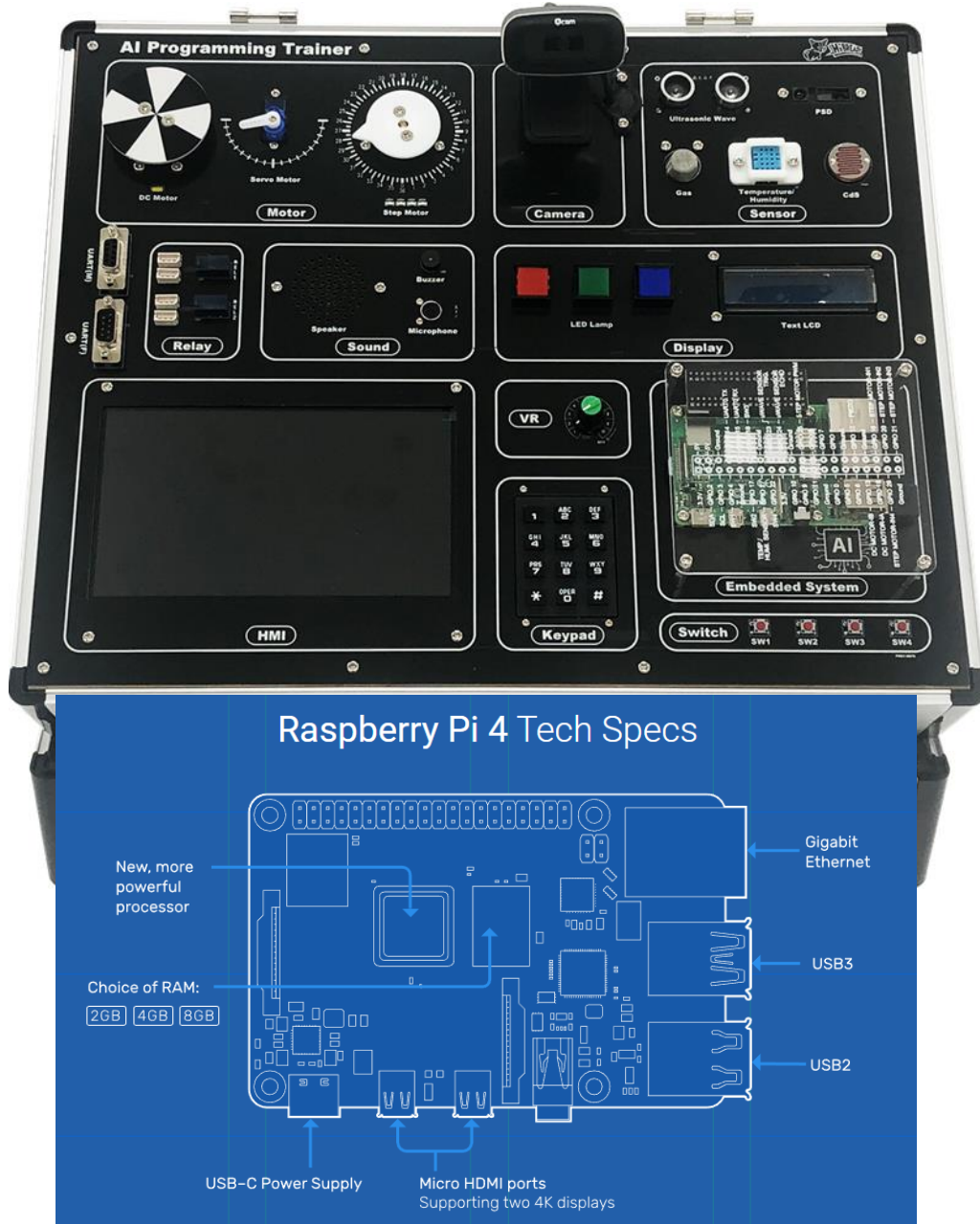


# 파이썬 활용

이영주  
young.kopo@gmail.com

# AI Programming



## Processor:

Broadcom BCM2711, quad-core Cortex-A72 (ARM v8)  
64-bit SoC @ 1.5GHz

## Memory:

1GB, 2GB, 4GB or 8GB LPDDR4  
(depending on model) with on-die ECC

## Connectivity:

2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless  
LAN, Bluetooth 5.0, BLE  
Gigabit Ethernet  
2 × USB 3.0 ports  
2 × USB 2.0 ports.

## GPIO:

Standard 40-pin GPIO header  
(fully backwards-compatible with previous boards)

## Video & sound:

2 × micro HDMI ports (up to 4Kp60 supported)  
2-lane MIPI DSI display port  
2-lane MIPI CSI camera port  
4-pole stereo audio and composite video port

## Multimedia:

H.265 (4Kp60 decode);  
H.264 (1080p60 decode, 1080p30 encode);  
OpenGL ES, 3.0 graphics

## SD card support:

Micro SD card slot for loading operating system  
and data storage

## Input power:

5V DC via USB-C connector (minimum 3A<sup>1</sup>)  
5V DC via GPIO header (minimum 3A<sup>1</sup>)  
Power over Ethernet (PoE)-enabled  
(requires separate PoE HAT)

## Environment:

Operating temperature 0–50°C

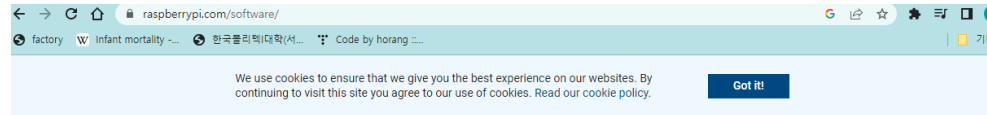
## Compliance:

For a full list of local and regional product approvals,  
please visit  
<https://www.raspberrypi.org/documentation/hardware/raspberrypi/conformity.md>

## Production lifetime:

The Raspberry Pi 4 Model B will remain in production  
until at least January 2026.

# 라즈베리파이 설치



## Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.



### Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

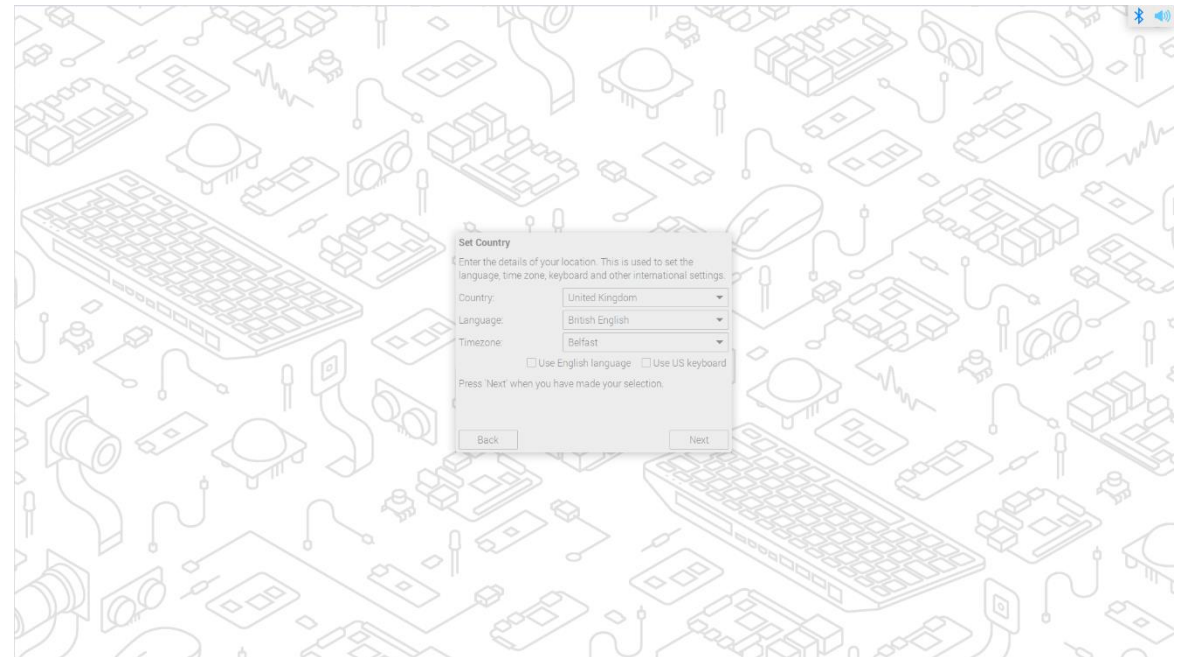
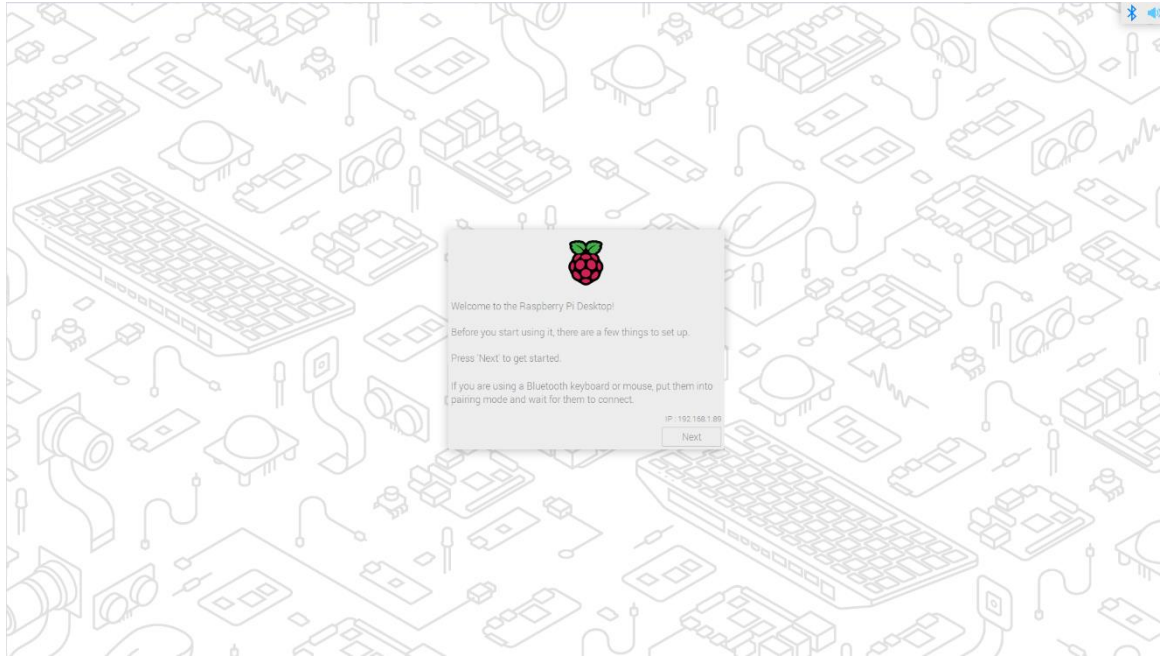
Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

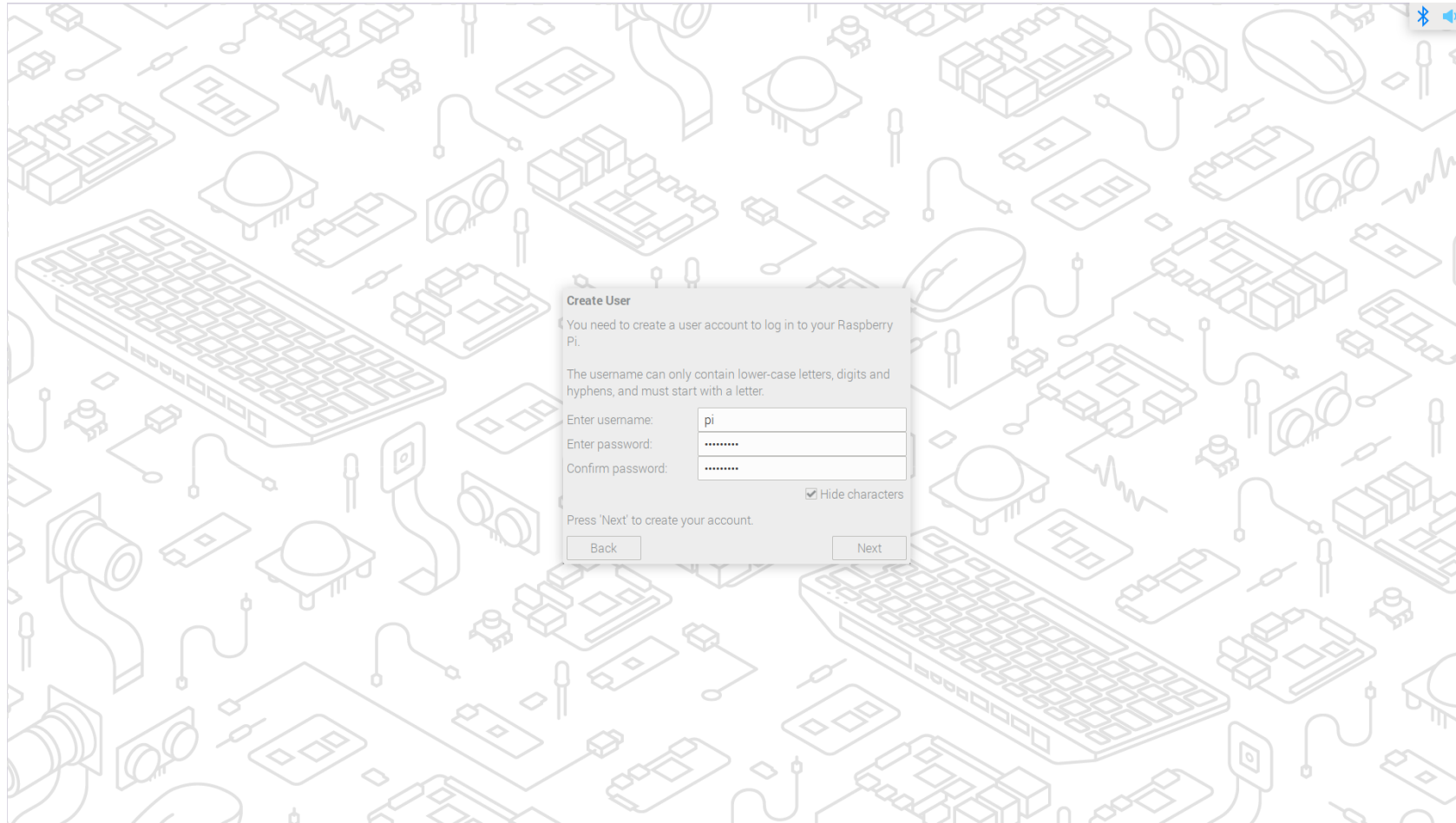
[Download for Windows](#)

[Download for macOS](#)



# OS 설치

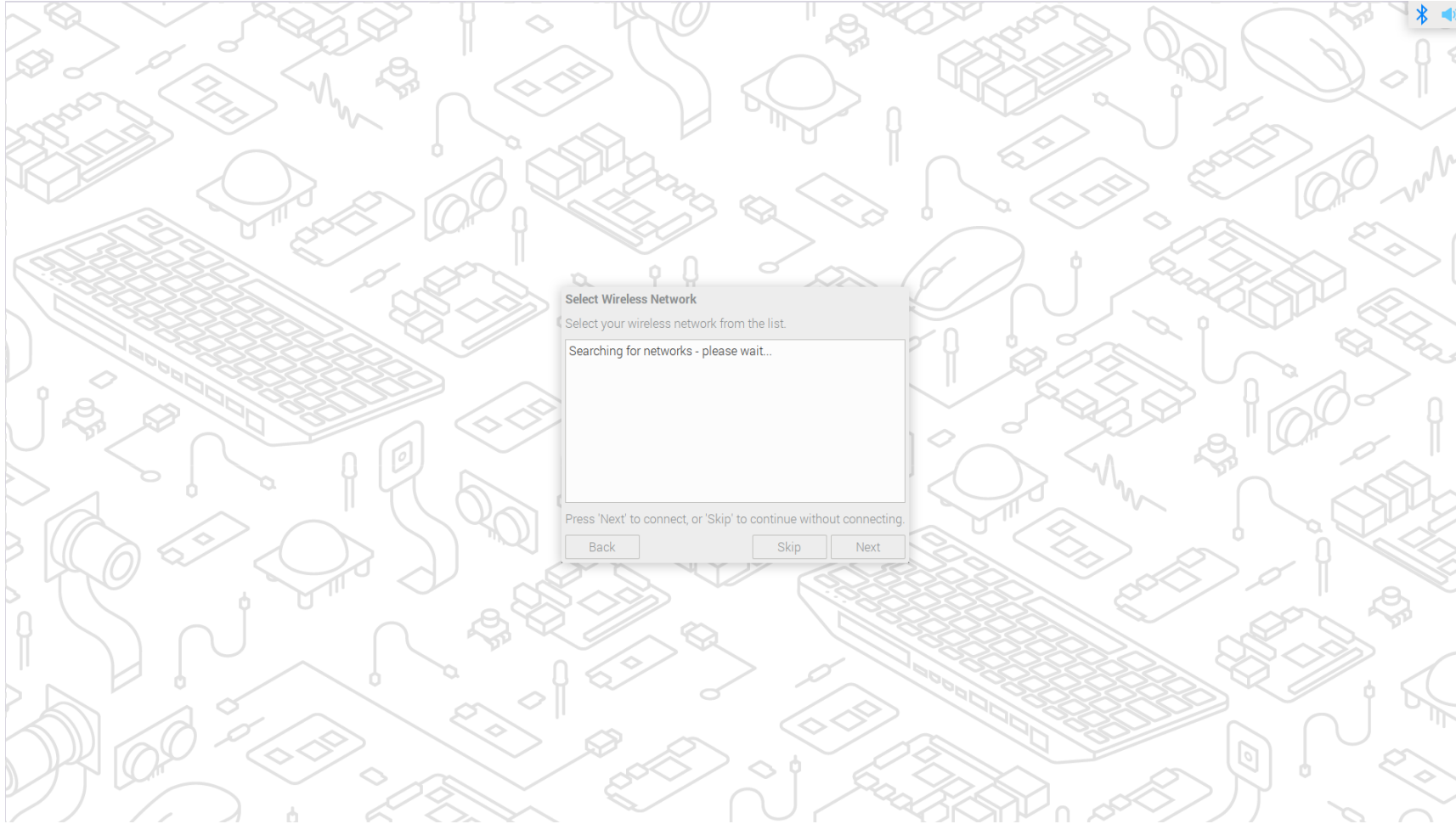




ID: pi  
Pwd:raspberry

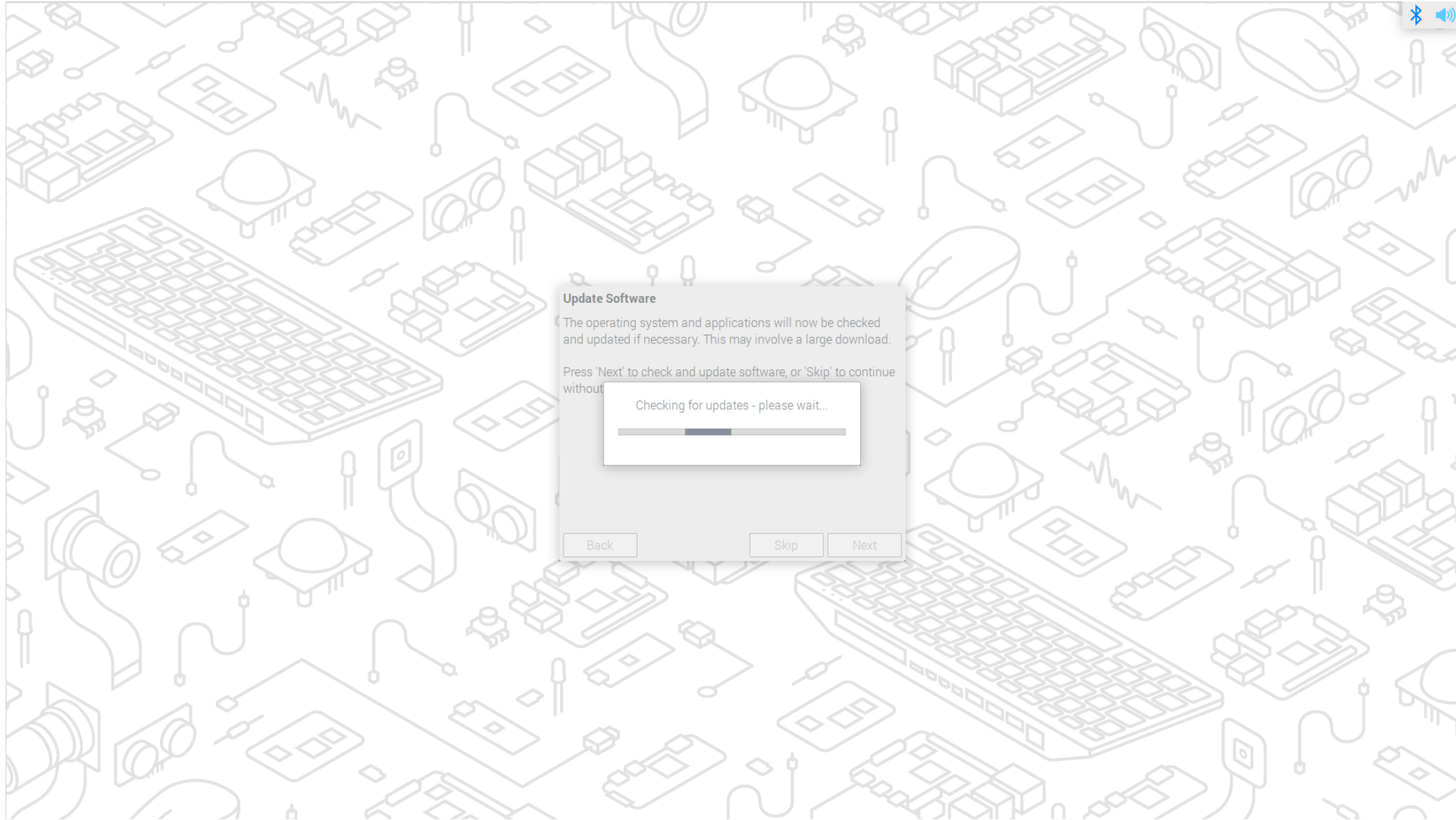


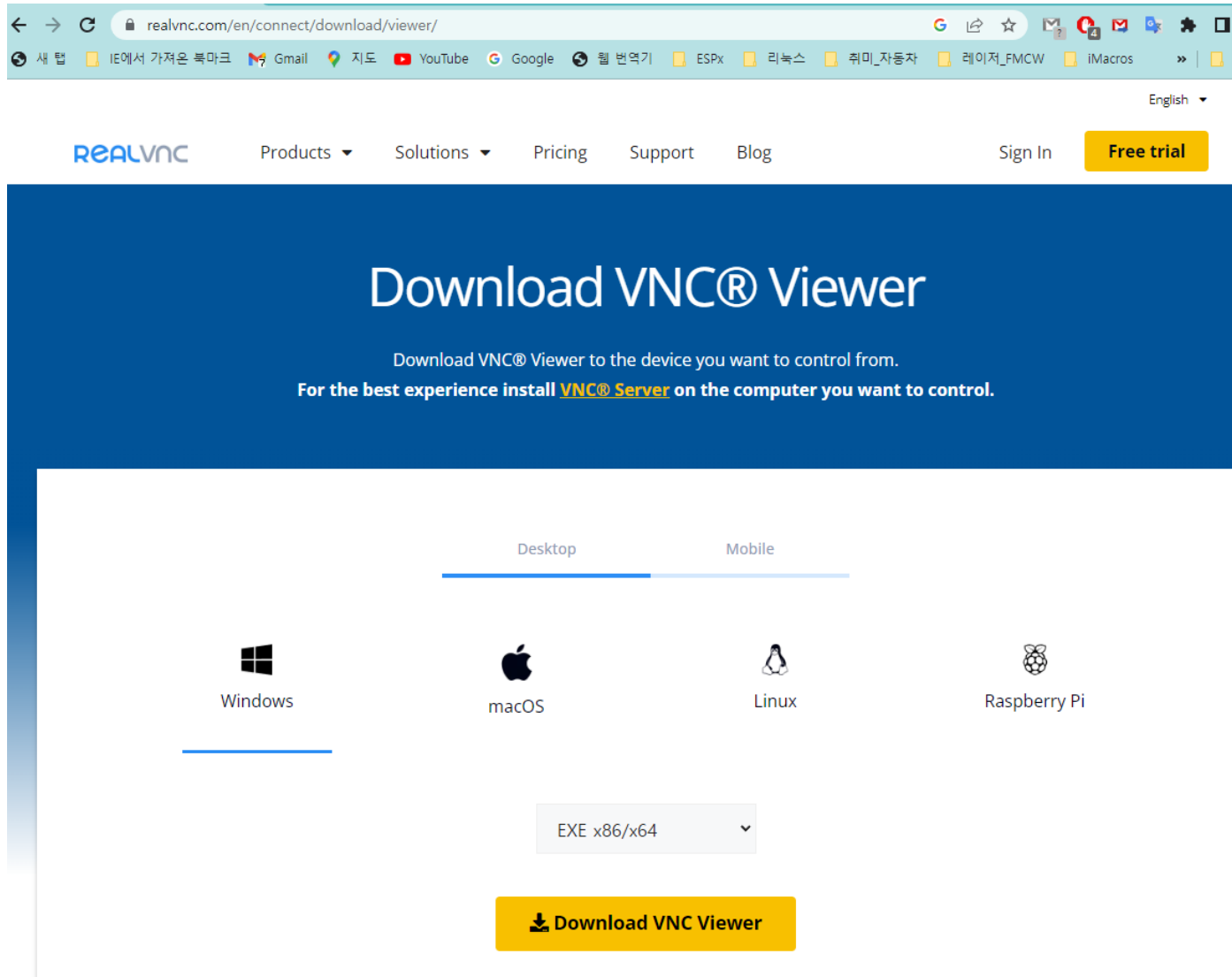
# 네트워크 설정- Wifi설정



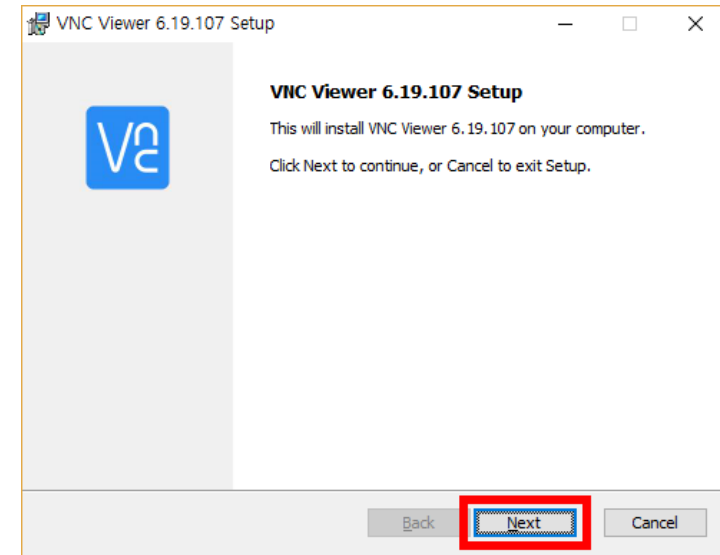
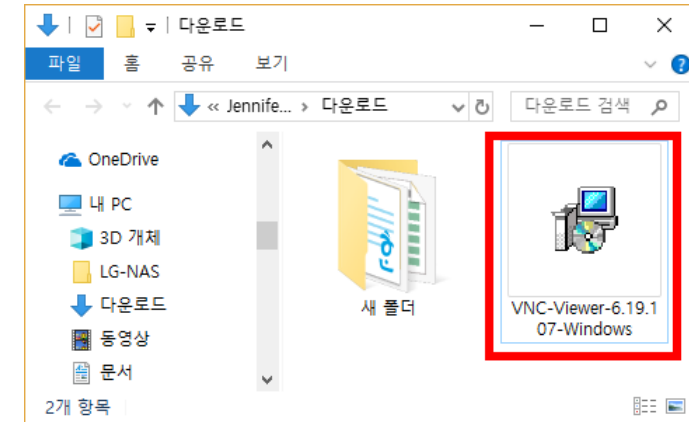
Ssid:AISW  
PWD:@Polytech

# 최신업데이트





The screenshot shows the RealVNC website's download page. The browser address bar displays `realvnc.com/en/connect/download/viewer/`. The navigation menu includes links for Products, Solutions, Pricing, Support, and Blog, along with a Sign In button and a Free trial button. The main heading is "Download VNC® Viewer". Below this, it states: "Download VNC® Viewer to the device you want to control from. For the best experience install **VNC® Server** on the computer you want to control." The page features tabs for Desktop and Mobile, with Desktop selected. Under the Desktop tab, there are icons for Windows, macOS, Linux, and Raspberry Pi. The Windows icon is highlighted. Below these icons is a dropdown menu showing "EXE x86/x64". At the bottom, there is a large yellow button labeled "Download VNC Viewer".





VNC Viewer

File View Help

vnc connect  
by RealVNC

192.168.25.47

Sign in...



192.168.25.47



192.168.25.139

Example

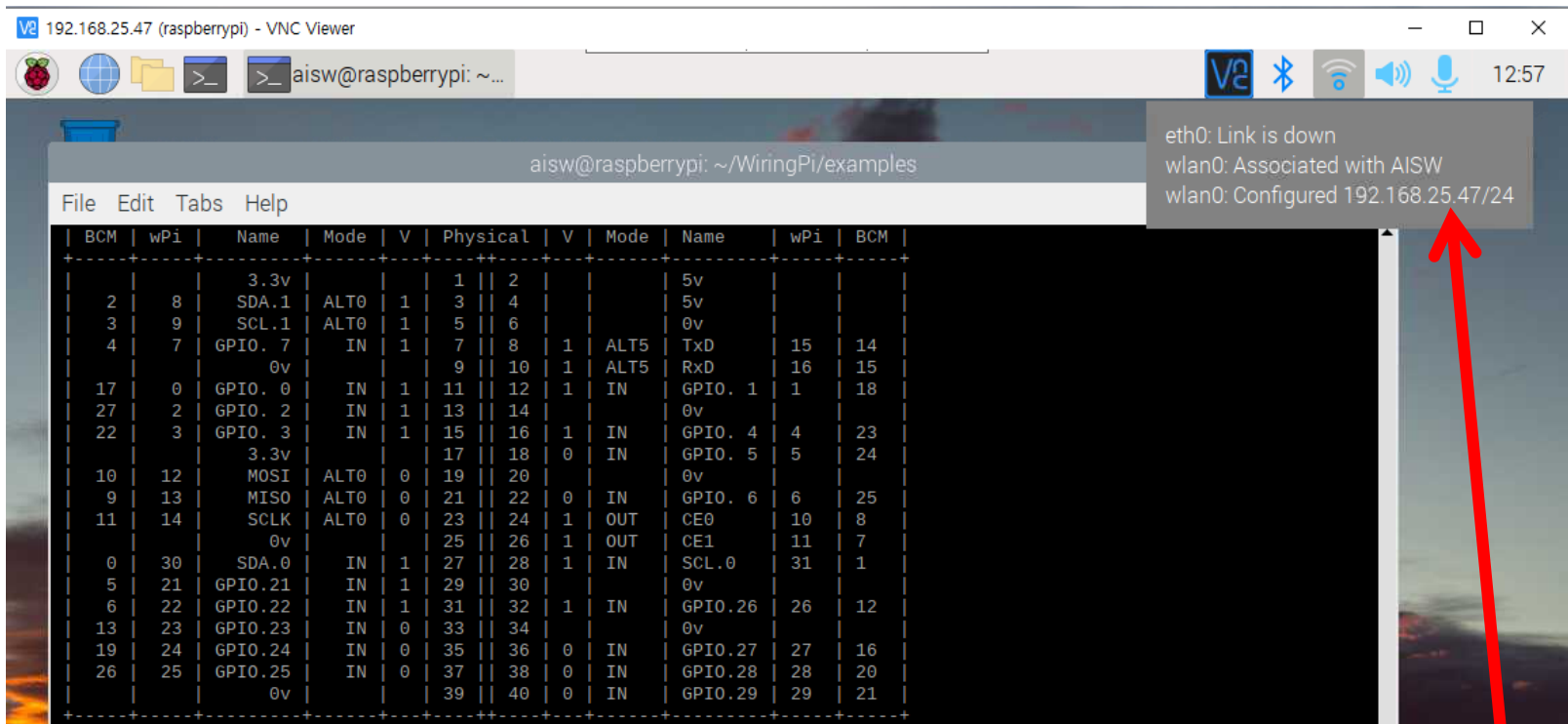
Workspace

TestSoftware

Thonny  
Python IDE

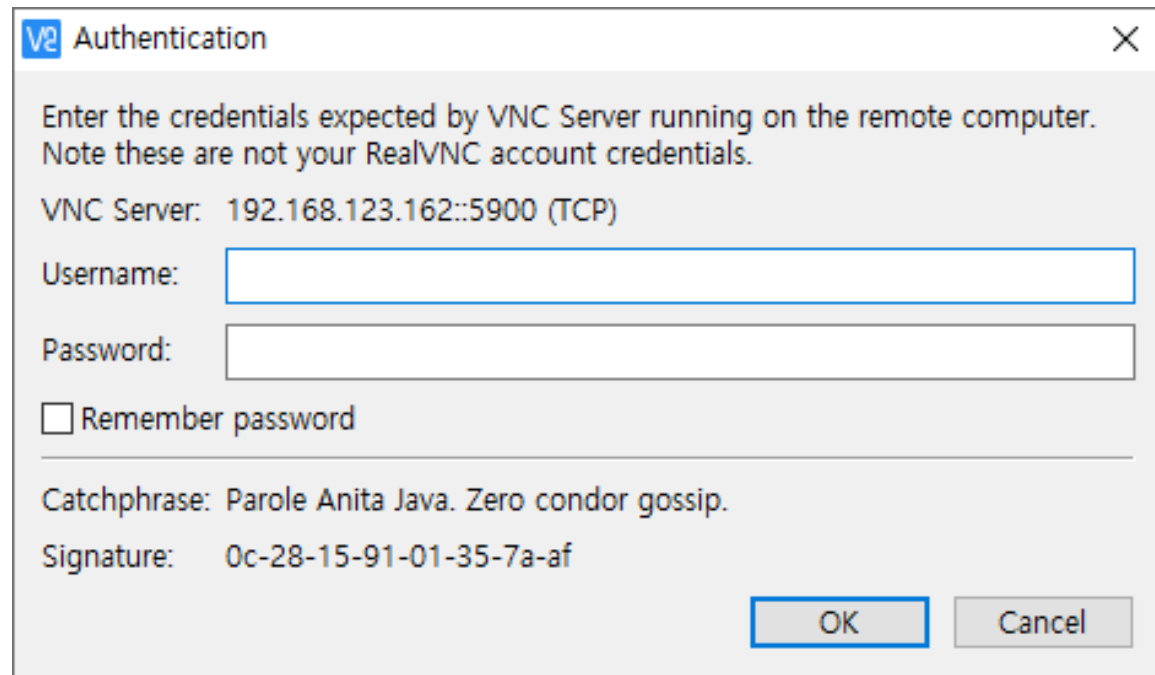
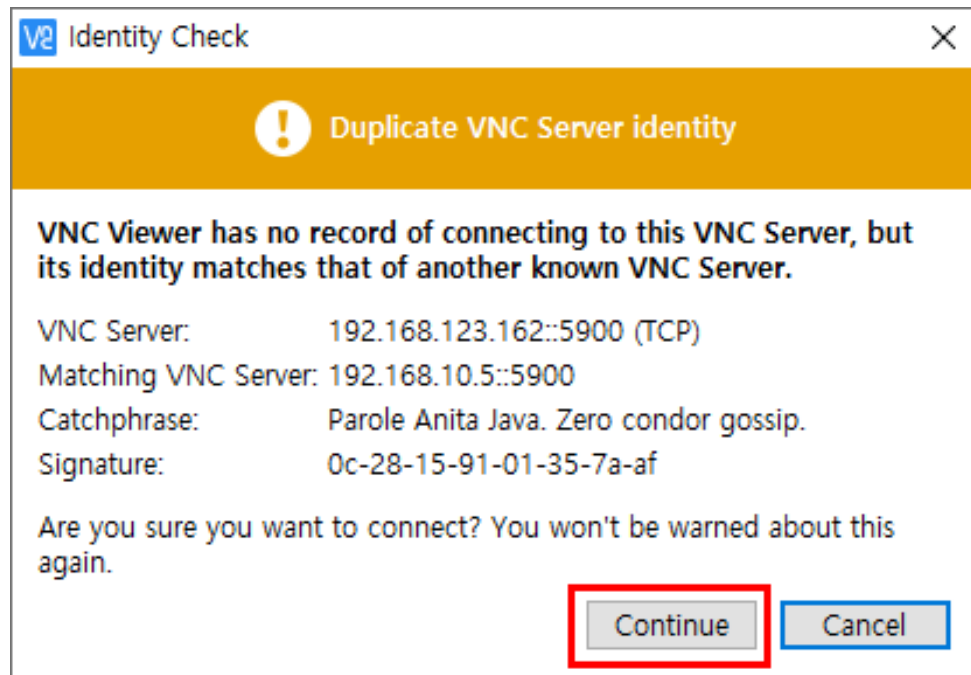
휴지통

10:19



접속IP입력





Username : bready

Password : 00000000

# C언어를 이용한 제어- 최신버전 wiringpi 설치

- 기존 버전 삭제: `sudo apt purge wiringpi`
- 최신버전 다운받기:
  - `git clone https://github.com/WiringPi/WiringPi.git`
- 확인 및 빌드
  - `cd WiringPi`
  - `sudo git pull origin`
  - `./build`

```
aisw@raspberrypi:~ $ hash -r
aisw@raspberrypi:~ $ git clone https://github.com/WiringPi/WiringPi.git
Cloning into 'WiringPi'...
remote: Enumerating objects: 1733, done.
remote: Counting objects: 100% (614/614), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 1733 (delta 553), reused 498 (delta 498), pack-reused 1119
Receiving objects: 100% (1733/1733), 803.80 KiB | 764.00 KiB/s, done.
Resolving deltas: 100% (1187/1187), done.
```

# Wiringpi 설치- 버전확인, gpio 포트확인

```
aisw@raspberrypi:~ $ cd WiringPi/
aisw@raspberrypi:~/WiringPi $ ls
build      debian-template  gpio      People    update    wiringPi
COPYING.LESSER  devLib          INSTALL  pins      VERSION   wiringPiD
debian      examples        newVersion  README.md  version.h
aisw@raspberrypi:~/WiringPi $ git pull origin
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only        # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
aisw@raspberrypi:~/WiringPi $ sudo git pull origin
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only        # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
Already up to date.
aisw@raspberrypi:~/WiringPi $ ./build
WiringPi Build script
```

```
aisw@raspberrypi:~/WiringPi $ gpio -v
gpio version: 2.70
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty
```

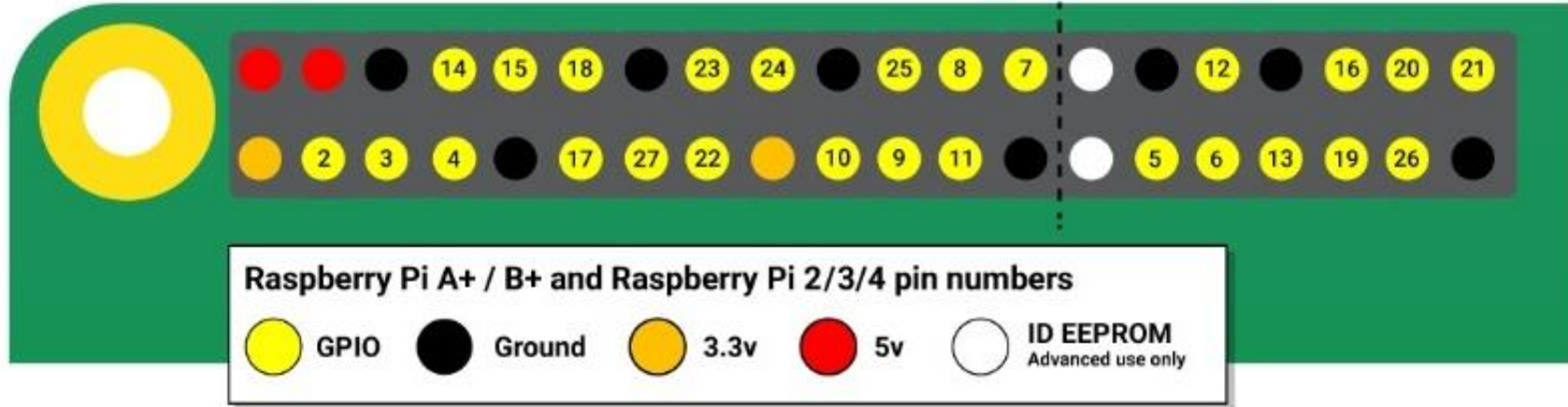
## Raspberry Pi Details:

```
Type: Pi 4B, Revision: 04, Memory: 4096MB, Maker: Sony
* Device tree is enabled.
*--> Raspberry Pi 4 Model B Rev 1.4
* This Raspberry Pi supports user-level GPIO access.
```

```
aisw@raspberrypi:~/WiringPi $ gpio readall
```

-----Pi 4B-----											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
-----											
		3.3v			1	2		5v			
2	8	SDA.1	ALT0	1	3	4		5v			
3	9	SCL.1	ALT0	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT5	TxD	15	14
		0v			9	10	1	ALT5	RxD	16	15
17	0	GPIO. 0	IN	1	11	12	1	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	1	13	14		0v			
22	3	GPIO. 3	IN	1	15	16	1	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19	20		0v			
9	13	MISO	ALT0	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23	24	1	OUT	CE0	10	8
		0v			25	26	1	OUT	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	1	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21
-----Pi 4B-----											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
-----											

GPIO는 General Purpose Input Output으로 범용 입/출력 인터페이스



▶ PWM(펄스 폭 변조)

- 소프트웨어 PWM : 모든 핀에서 설정 가능
- 하드웨어 PWM : GPIO12, GPIO13, GPIO18, GPIO19에서 사용 가능

▶ SPI

- SPI0 : MOSI(GPIO10), MISO(GPIO9), SCLK(GPIO11), CE0(GPIO8), CE1(GPIO7)
- SPI1 : MOSI(GPIO20), MISO(GPIO19), SCLK(GPIO21), CE0(GPIO18), CE1(GPIO17), CE2(GPIO16)

▶ I2C

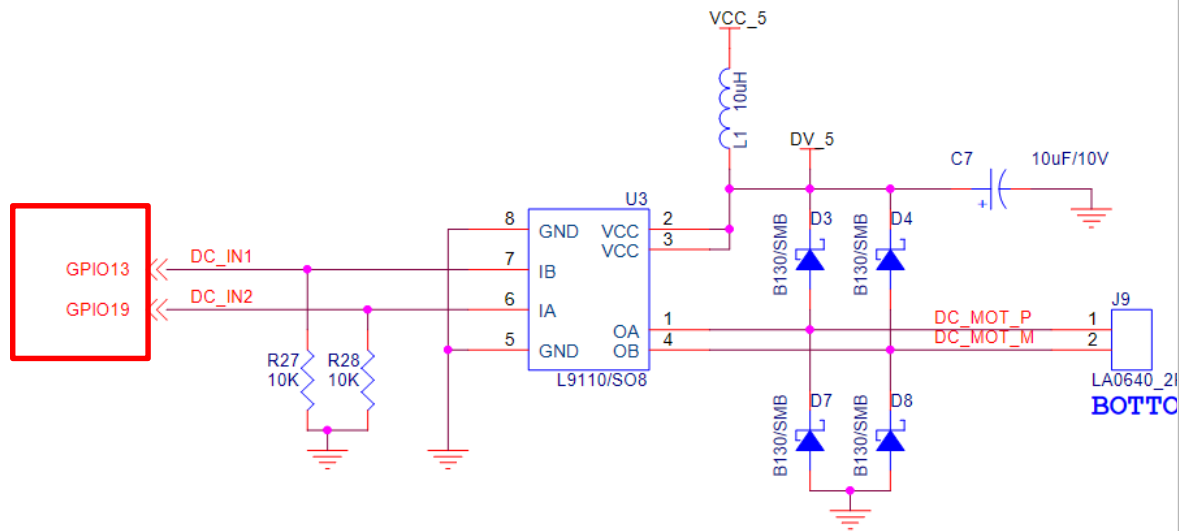
- Data(GPIO2), Clock(GPIO3)
- EEPROM Data(GPIO0), EEPROM Clock(GPIO1)

▶ Serial

- Tx(GPIO14), Rx(GPIO15)



# 모터제어를 위한 GPIO설정



BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
2	8	3.3v			1	2		5v		
3	9	SDA.1	ALT0	1	3	4		5v		
4	7	SCL.1	ALT0	1	5	6		0v		
		GPIO. 7	IN	1	7	8	1	ALT5	TxD	15
		0v			9	10	1	ALT5	RxD	16
17	0	GPIO. 0	IN	1	11	12	1	IN	GPIO. 1	1
27	2	GPIO. 2	IN	1	13	14		0v		
22	3	GPIO. 3	IN	1	15	16	1	IN	GPIO. 4	4
		3.3v			17	18	0	IN	GPIO. 5	5
10	12	MOSI	ALT0	0	19	20		0v		
9	13	MISO	ALT0	0	21	22	0	IN	GPIO. 6	6
11	14	SCLK	ALT0	0	23	24	1	OUT	CE0	10
		0v			25	26	1	OUT	CE1	11
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31
5	21	GPIO.21	IN	1	29	30		0v		
6	22	GPIO.22	IN	1	31	32	1	IN	GPIO.26	26
13	23	GPIO.23	IN	0	33	34		0v		
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28
		0v			39	40	0	IN	GPIO.29	29

BCM GPIO13 → WiringPi GPIO23  
 BCM GPIO19 → WiringPi GPIO24

# 모터제어

```
aisw@raspberrypi:~/WiringPi/examples $ sudo nano motor.c
```

```
#include <stdio.h>
#include <wiringPi.h>

#define motor_pin_n 23
#define motor_pin_p 24

int main(){

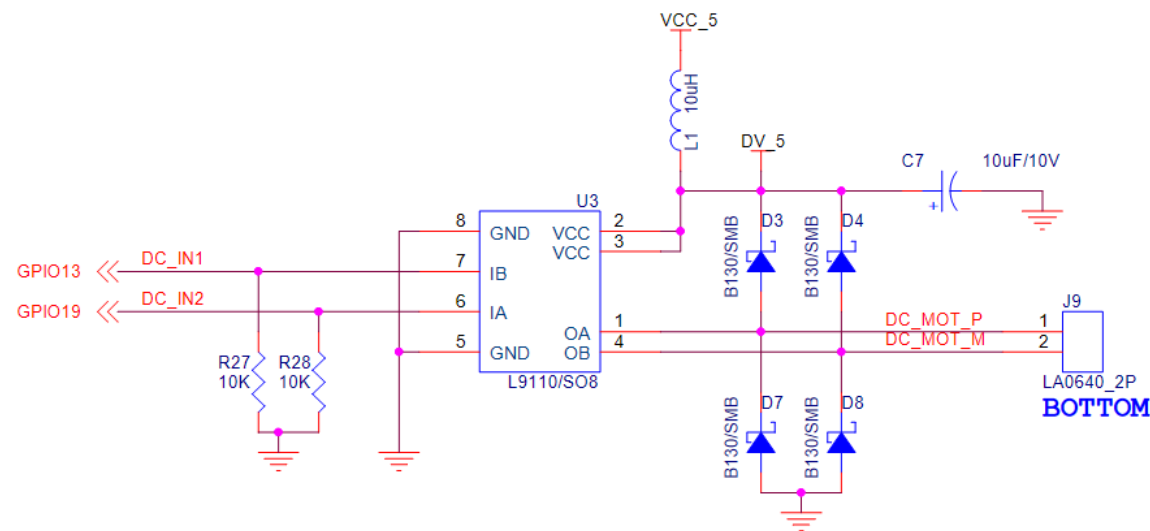
    wiringPiSetup();

    pinMode(motor_pin_n, OUTPUT);
    pinMode(motor_pin_p, OUTPUT);

    digitalWrite(motor_pin_n, LOW);
    digitalWrite(motor_pin_p, HIGH);
    delay(500);

    digitalWrite(motor_pin_n, LOW);
    digitalWrite(motor_pin_p, LOW);
    printf("Motor Test");

    return 0;
}
```




BCM GPIO13 → WiringPi GPIO23  
BCM GPIO19 → WiringPi GPIO24

```
aisw@raspberrypi:~/WiringPi/examples $ sudo gcc -o motor motor.c -lwiringPi
```

```
aisw@raspberrypi:~/WiringPi/examples $ ./motor
```

## 라즈베리파이 GPIO 제어용 파이썬 모듈 (기본 내장)



[Help](#) [Sponsors](#) [Log in](#) [Register](#)

# RPi.GPIO 0.7.1

`pip install RPi.GPIO`

 [Latest version](#)

Released: Feb 7, 2022

A module to control Raspberry Pi GPIO channels

### Navigation

- Project description**
-  Release history
-  Download files

---

### Project links

-  [Homepage](#)

---

### Project description

This package provides a Python module to control the GPIO on a Raspberry Pi.

Note that this module is unsuitable for real-time or timing critical applications. This is because you can not predict when Python will be busy garbage collecting. It also runs under the Linux kernel which is not suitable for real time applications - it is multitasking O/S and another process may be given priority over the CPU, causing jitter in your program. If you are after true real-time performance and predictability, buy yourself an Arduino <http://www.arduino.cc> !

Note that the current release does not support SPI, I2C, hardware PWM or serial functionality on the RPi yet. This is planned for the near future - watch this space! One-wire functionality is also planned.

Although hardware PWM is not available yet, software PWM is available to use on all channels.

For examples and documentation, visit <http://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>

## import RPi.GPIO as GPIO



The screenshot shows the SourceForge website for the 'raspberrypi-python' project. The page is titled 'raspberrypi-python Wiki' and describes it as 'A Python module to control the GPIO on a Raspberry Pi'. The 'Wiki' tab is selected in the navigation bar. The left sidebar contains a search bar and links to 'Wiki Home', 'Browse Pages', 'Browse Labels', and 'Formatting Help'. The main content area is titled 'BasicUsage' and shows the 'Authors: Anonymous' section. The main heading is 'RPi.GPIO module basics' followed by 'Importing the module'. The text explains how to import the RPi.GPIO module and provides a code snippet for a try-except block to handle potential runtime errors.

SourceForge

Open Source Software Business Software Resources

Help Create Join Login

Sync your GitHub Project to SourceForge

Search for software or solutions

Home / Browse / raspberrypi-python / Wiki

**raspberrypi-python Wiki**  
A Python module to control the GPIO on a Raspberry Pi  
Brought to you by: [croston](#)

Summary Files Reviews Support Tickets Wiki Code

Search Wiki

Wiki Home  
Browse Pages  
Browse Labels  
Formatting Help

**BasicUsage**

Authors: Anonymous

### RPi.GPIO module basics

#### Importing the module

To import the RPi.GPIO module:

```
import RPi.GPIO as GPIO
```

By doing it this way, you can refer to it as just GPIO through the rest of your script.

To import the module and check to see if it is successful:

```
try:
    import RPi.GPIO as GPIO
except RuntimeError:
    print("Error importing RPi.GPIO! This is probably because you need superuser privileges. You can achieve this by using 'sudo' to run your script")
```

- 제어핀 모드 설정

```
GPIO.setmode(GPIO.BOARD)
# or
GPIO.setmode(GPIO.BCM)
```



		Physical Pins			
Function	BCM	pin#	pin#	BCM	Function
3.3 Volts		1	2		5 Volts
GPIO/SDA1 (I2C)	2	3	4		5 Volts
GPIO/SCL1 (I2C)	3	5	6		GND
GPIO/GCLK	4	7	8	14	TX UART/GPIO
GND		9	10	15	RX UART/GPIO
GPIO	17	11	12	18	GPIO
GPIO	27	13	14		GND
GPIO	22	15	16	23	GPIO
3.3 Volts		17	18	24	GPIO
MOSI (SPI)	10	19	20		GND
MISO(SPI)	9	21	22	25	GPIO
SCLK(SPI)	11	23	24	8	CEO_N (SPI)
GND		25	26	7	CE1_N (SPI)
RESERVED		27	28		RESERVED
GPIO	5	29	30		GND
GPIO	6	31	32	12	GPIO
GPIO	13	33	34		GND
GPIO	19	35	36	16	GPIO
GPIO	26	37	38	20	GPIO
GND		39	40	21	GPIO

- 채널 설정(입/출력 설정)
  - ✓ `GPIO.setup(channel, GPIO.IN)`
  - ✓ `GPIO.setup(channel, GPIO.OUT)`
  - ✓ `GPIO.setup(channel, GPIO.OUT, initial = GPIO.HIGH)`
- 입출력 제어
  - ✓ `GPIO.input(channel)`
  - ✓ `GPIO.output(channel, state)`
  - state : `GPIO.HIGH/1/True`, `GPIO.LOW/0/False`
- 종료전 리소스 반납(필수)
  - ✓ `GPIO.cleanup()`



# Rpi.GPIO 사용 예제

```
# 라이브러리 импорт
import RPi.GPIO as GPIO

...
# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(12, GPIO.IN)
GPIO.setup(18, GPIO.OUT)

...
# 메인 쓰레드
try:
    while 1:
        button = GPIO.input(12)
        ...
        GPIO.output(18, GPIO.HIGH)
        ...

# 반드시 클린업
finally: GPIO.cleanup()
```